

CLAIMS

What is claimed is:

1. A method for performing a pattern match search for a data string having a plurality of characters separated by delimiters, said method comprising:

defining a first category of characters as delimiters such that all remaining characters are defined as non-delimiters;

constructing a search key by:

generating a full match search increment comprising the binary representation of a data string element, wherein said data string element comprises all non-delimiters between a pair of said delimiters; and

concatenating a pattern search prefix to said full match search increment to form said search key, wherein said pattern search prefix is a cumulative pattern search result of each previous full match search increment;

performing a full match search within a lookup table utilizing said search key;

in response to finding a matching pattern within said lookup table, returning to said step of constructing a search key; and

in response to not finding a matching pattern, utilizing the previous full match search result to process said data string.

1 2. The method of claim 1, wherein said step of constructing a search key is
2 preceded by pointing to a character within said data string.

1 3. The method of claim 2, wherein said step of constructing a search key further
2 comprises:

3 evaluating said character to determine whether or not said character is a
4 delimiter;

5 in response to said character being a delimiter:

6 delivering a full match search increment into a search key register, wherein
7 said search increment comprises a binary representation of all non-delimiters between
8 said delimiter and an immediately preceding delimiter; and

9 concatenating said pattern search prefix to said search increment within said
10 search key element;

11 in response to said character not being a delimiter, appending a binary
12 representation of said character to said search increment; and

13 incrementing said pointer.

1 4. The method of claim 1, further comprising in response to finding a matching
2 pattern, updating said pattern search prefix.

1 5. The method of claim 1, wherein said step of performing a full match search
2 further comprises:

3 determining whether or not a full match for said search key exists within said
4 hash table by:

5 hashing said search key to produce a hash key result;

6 indexing a hash table utilizing said hash key result to find a matching stored
7 pattern; and

8 resolving collisions in said hash table utilizing a pattern search control block.

1 6. The method of claim 1, wherein said data string is a Universal Resource
2 Indicator address, and wherein said data string element is a URI element.

3 7. The method of claim 6, wherein said delimiters comprise period characters or
4 slash characters.

5 8. The method of claim 6, wherein said step of constructing a search key is
6 preceded by the steps of:

7 scanning an IP data packet to determine a first URI element to be parsed;

8 initializing a URI pointer to a first character within said first URI element; and

9 initializing said pattern search prefix to zero.

1 9. A system for performing a pattern match search for a data string having a
2 plurality of characters separated by delimiters, said system comprising:

3 means for defining a first category of characters as delimiters such that all
4 remaining characters are defined as non-delimiters;

5 processing means for constructing a search key by:

6 generating a full match search increment comprising the binary representation
7 of a data string element, wherein said data string element comprises all non-delimiters
8 between a pair of said delimiters; and

9 concatenating a pattern search prefix to said full match search increment to
10 form said search key, wherein said pattern search prefix is a cumulative pattern search
11 result of each previous full match search increment;

12 processing means for performing a full match search within a lookup table
13 utilizing said search key;

14 processing means response to finding a matching pattern within said lookup
15 table for returning to said step of constructing a search key; and

16 processing means responsive to not finding a matching pattern for utilizing the
17 previous full match search result to process said data string.

1 10. The system of claim 9, further comprising processing means for pointing to a
2 character within said data string prior to constructing a search key.

1 11. The system of claim 10, wherein said processing means for constructing a
2 search key further comprises:

3 processing means for evaluating said character to determine whether or not
4 said character is a delimiter;

5 processing means responsive to said character being a delimiter for:

6 delivering a full match search increment into a search key register, wherein
7 said search increment comprises a binary representation of all non-delimiters between
8 said delimiter and an immediately preceding delimiter; and

9 concatenating said pattern search prefix to said search increment within said
10 search key element;

11 processing means responsive to said character not being a delimiter for
12 appending a binary representation of said character to said search increment; and

13 processing means for incrementing said pointer.

1 12. The system of claim 9, further comprising processing means responsive to
2 finding a matching pattern for updating said pattern search prefix.

1 13. The system of claim 9, wherein said processing means for performing a full
2 match search further comprises:

3 processing means for determining whether or not a full match for said search
4 key exists within said hash table by:

5 hashing said search key to produce a hash key result;

6 indexing a hash table utilizing said hash key result to find a matching stored
7 pattern; and

8 resolving collisions in said hash table utilizing a pattern search control block.

1 14. The system of claim 9, wherein said data string is a Universal Resource
2 Indicator address, and wherein said data string element is a URI element.

3 15. The system of claim 14, wherein said delimiters comprise period characters or
4 slash characters.

5 16. The system of claim 14, wherein said processing means for constructing a
6 search key further comprises:

7 processing means for scanning an IP data packet to determine a first URI
8 element to be parsed;

9 processing means for initializing a URI pointer to a first character within said
10 first URI element; and

11 processing means for initializing said pattern search prefix to zero.

1 17. A computer program product for performing a pattern match search for a data
2 string having a plurality of characters separated by delimiters, said computer program
3 product comprising:

4 instruction means for defining a first category of characters as delimiters such
5 that all remaining characters are defined as non-delimiters;

6 instruction means for constructing a search key by:

7 generating a full match search increment comprising the binary representation
8 of a data string element, wherein said data string element comprises all non-delimiters
9 between a pair of said delimiters; and
10

11 concatenating a pattern search prefix to said full match search increment to
12 form said search key, wherein said pattern search prefix is a cumulative pattern search
13 result of each previous full match search increment;

14 instruction means for performing a full match search within a lookup table
15 utilizing said search key;

16 instruction means response to finding a matching pattern within said lookup
17 table for returning to said step of constructing a search key; and

18 instruction means responsive to not finding a matching pattern for utilizing the
previous full match search result to process said data string.

1 18. The computer program product of claim 17, further comprising instruction
2 means for pointing to a character within said data string prior to constructing a search
3 key.

1 19. The computer program product of claim 18, wherein said instruction means
2 for constructing a search key further comprises:

3 instruction means for evaluating said character to determine whether or not
4 said character is a delimiter;

5 instruction means responsive to said character being a delimiter for:

6 delivering a full match search increment into a search key register, wherein
7 said search increment comprises a binary representation of all non-delimiters between
8 said delimiter and an immediately preceding delimiter; and

9 concatenating said pattern search prefix to said search increment within said
10 search key element;

11 instruction means responsive to said character not being a delimiter for
12 appending a binary representation of said character to said search increment; and

13 instruction means for incrementing said pointer.

1 20. The computer program product of claim 17, further comprising instruction
2 means responsive to finding a matching pattern for updating said pattern search
3 prefix.

1 21. The computer program product of claim 71, wherein said instruction means
2 for performing a full match search further comprises:

3 instruction means for determining whether or not a full match for said search
4 key exists within said hash table by:

5 hashing said search key to produce a hash key result;

6 indexing a hash table utilizing said hash key result to find a matching stored
7 pattern; and

8 resolving collisions in said hash table utilizing a pattern search control block.

22. The computer program product of claim 17, wherein said data string is a
Universal Resource Indicator address, and wherein said data string element is a URI
element.

23. The computer program product of claim 22, wherein said delimiters comprise
period characters or slash characters.

24. The computer program product of claim 22, wherein said instruction means
for constructing a search key further comprises:

3 instruction means for scanning an IP data packet to determine a first URI
4 element to be parsed;

5 instruction means for initializing a URI pointer to a first character within said
6 first URI element; and

7 instruction means for initializing said pattern search prefix to zero.